

---

---

# Coding Dojo experience

Prof. Missiroli

---

“Acquiring coding skills should be a continuous process...”

<http://codingdojo.org/>

# Dōjō (道場)

Luogo in cui si praticano le arti marziali. Trasposto nel campo informatico a partire dal 2004 a Parigi.

In entrambi i casi:

- Un luogo “sicuro” dove fare pratica.
- Un luogo dove fare errori ed imparare.
- Ci si riunisce per un Kata, ci si diverte e si fa [pratica deliberata](#).



# Coding Dōjō

Obiettivi:

- Fare pratica.
- Imparare nuove tecniche.
- Migliorare tecniche esistenti
- NON NECESSARIAMENTE risolvere il problema



# Coding Dōjō

Valori:

- Ambiente sicuro
- Nuove idee apprezzate.
- Collaborativo e non competitivo
- Adatto a tutti i livelli di difficoltà

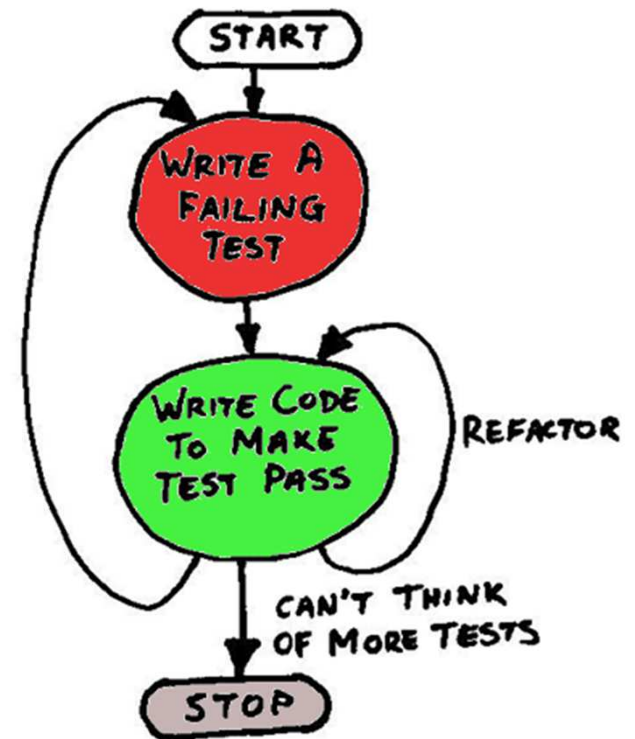




# Coding Dōjō

Struttura:

- Un computer con proiettore
- Test Driven Development
- Piccoli incrementi ("Baby steps")
- Tutti devono capire
- SEMPRE partire da zero



# Regole: NON...

**..saltare alle conclusioni.**

Piccoli passi, no?

**...lanciarsi in sterili discussioni.**

Il tempo è prezioso

**...combattere con gli altri partecipanti**

Si collabora, non si compete

# Etica

Se sai scrivere il codice, vieni a scriverlo.

Se scrivi il codice, tutti devono capire cosa stai facendo.

Se sei il prossimo a scrivere il codice, non interrompere il flusso.

Se non stai scrivendo codice, non disturbare.

Se hai un'idea, mostrane il codice.

Se sei in difficoltà chiedi aiuto.





# 1. Kata Preparato

Rinfrescare, chiarire, esporre nuovi concetti.i.

→ **Presentazione**

Un "esperto" mostra come risolvere il problema

→ **Si parte da zero**

→ **Usare TDD & Baby steps**

→ **Tutti devono capire ogni passo**

→ **Interrompere se non si capisce  
che sta succedendo o ci si è  
persi**

# Il problema: FizzBuzz

Hand-drawn diagram illustrating the FizzBuzz problem. The numbers 1 through 25 are arranged in a spiral pattern. Multiples of 3 are circled in green, and multiples of 5 are circled in red. The word "BUZZ" is written in red above the number 5, and the word "FIZZ" is written in green below the number 24.

Number	Divisible by 3	Divisible by 5	Output
1	No	No	1
2	No	No	2
3	Yes	No	3
4	No	No	4
5	No	Yes	5
6	Yes	No	6
7	No	No	7
8	No	No	8
9	Yes	No	9
10	No	Yes	10
11	No	No	11
12	Yes	No	12
13	No	No	13
14	No	No	14
15	Yes	Yes	15
16	No	No	16
17	No	No	17
18	Yes	No	18
19	No	No	19
20	No	Yes	20
21	Yes	No	21
22	No	No	22
23	No	No	23
24	Yes	No	24
25	No	Yes	25

---

# Step 1: Foundation

---

## Step 2: First test

---

## Step 3: Second test

---

## Step 3: Third test

---

## Step 4: Refactor?



—  
**Step n:** test, test, refactor

**SODDISFATTI?**

# Piccola Pausa

Forse!





## 2. Randori Kata

Esperienza di sviluppo condivisa

→ **Pair Programming**

Pilota e Navigatore.

→ **TDD+Babysteps**

Navigatore scrive il test, il pilota risolve.  
Si commenta pubblicamente quello  
che si fa.

→ **Timeslot MAX 5 minuti**

Al termine: il pilota diventa navigatore;  
un nuovo pilota arriva dalla platea.

# Scegliamo l'argomento

- Punteggio del Bowling
- Gioco della vita (Un po' lunghetto..)
- Conversione da intero a numero romano (59 -> LIX)
- Valutatore di una stringa



—

# Chi comincia?

Feedback time

QUI



# Ringraziamento

Gran parte delle idee sono prese e reinterpretate dalle slide "Coding Dojo instructions" di Viet Nguyen